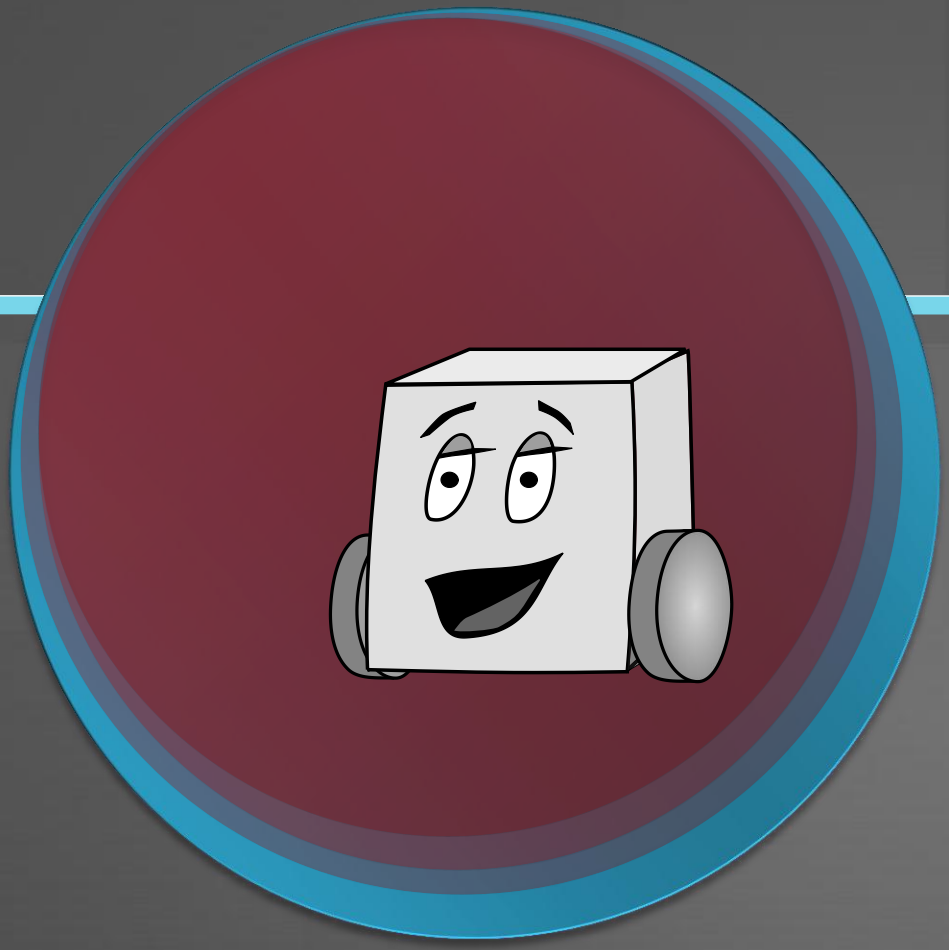


E11 Lecture 17: Debugging

Profs. David Money Harris & Sarah Harris

Fall 2011


DEBUGGING
RULES!
Understand the system
Make it fail
Quit thinking and look
Divide and conquer
Change one thing at a time
Keep an audit trail
Check the plug
Get a fresh view
If you didn't fix it, it ain't fixed
DEBUGGING
Rules
FIND BUGS FAST!
from Debugging © 2002 by David Agans
To get the book or download this free poster, go to
www.debuggingrules.com

Understand the System

- If you wrote it:
  - Do you understand how the language works?
  - Do you understand what the functions do?
  - Do you understand how it interfaces with the hardware?
- If somebody else wrote it?
  - What is it supposed to do?
  - How is it supposed to work?
  - Plus all of the above…
- Sometimes, a careful reading of the code reveals the bug

4 KHz Phototransistor Reading

```
#define PHOTOTRANS 4


for (i=0; i<31; i++) {
      beacon[i] = analogRead(PHOTOTRANS);
      delayMicroseconds(250);
}
```

Make it Fail

- Find a way to consistently reproduce the bug
  - If you can, you are halfway to finding the bug
  - If not, you have a very challenging bug on your hands
- If the bug is erratic:
  - Does it behave differently on your partner's board?
  - Does it behave differently based on the power or motors?
  - If it is a hardware failure, is it sensitive to V_{DD} or temperature?

Sporadic Spontaneous Failure

- Mudduino 1.0 would occasionally spontaneously freeze up for a second and then start over from the beginning of startup()
- Root Cause:

  A power surge from the motors caused the supply voltage to droop excessively and the Atmega reset itself.
- Fix:

  Add dedicated layers to the Mudduino PCB for power and ground. Add capacitors to stabilize power supply.

Quit Thinking and Look

- If you don't see the bug right away by inspection, start running the program.
- Predict what the program should do at each step (why?)
- Monitor what the program actually does
  - Print statements
  - Or a debugger
- Look for the first discrepancy
  - Now you've isolate the bug

Gold Code Correlation

```
if (abs(cor) > maxCor)) maxCor = cor;
```

- Results

```
Correlation GC 1, GC 4
-1 7 7 7 7 -1 7 7 -9 -9 -9 -9 -1 -9 -9 -1 7 -1 -1 7 7 7 -1 -1 -1 -9 -1 -1 -1 -1 -1
MAX: -1
```

- Add print statement:

```
if (abs(cor) > maxCor)) {
      maxCor = cor;
      Serial.print("Biggest so far: ");
      Serial.println(maxCor);
}
```

Divide and Conquer

- Searching for the bug in a big program line by line takes too long.
- Look at the results in the middle.
  - If they are good, the problem is later.
  - If they are bad, the problem is earlier
  - Recursively search in first or second half of the program
- Called a *binary search*

Change One Thing at a Time

- If you make multiple changes at a time, it is hard to know which one fixed the problem.
- Sometimes you introduce a new bug at the same time you fix the current one.

Keep an Audit Trail

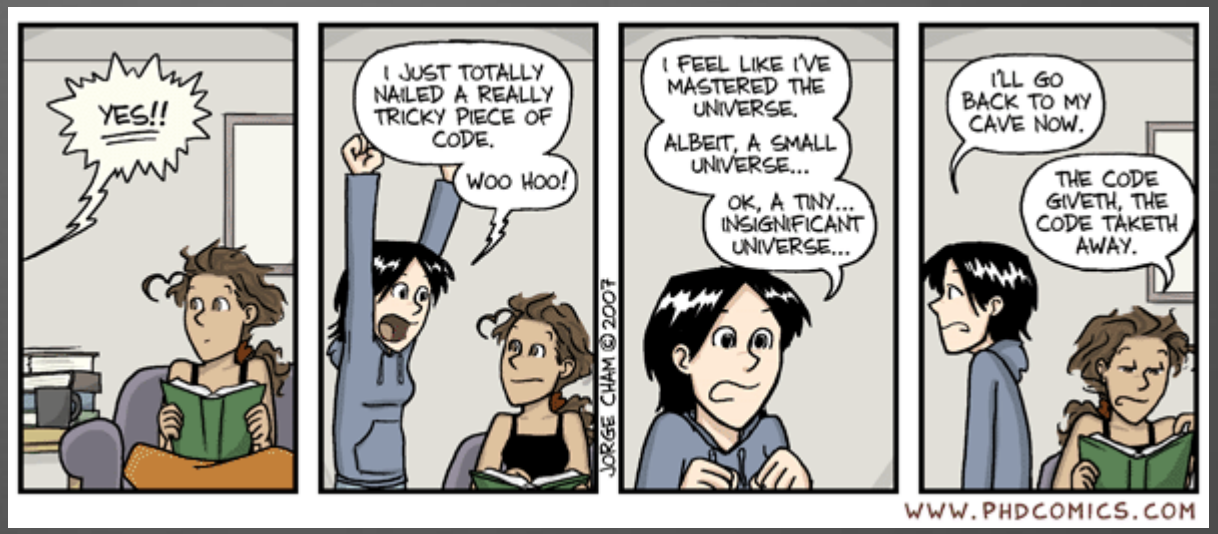
- It is easy to forget what tests you have done
  - Especially when the problem is erratic
- Be methodical
- Keep a lab notebook
- For each attempted bug fix, record
  - Hypothesis of the cause
  - Description of how you are trying to fix it
  - Outcome of the attempt

Check the Plug

- Easy to overlook "obvious" problems

Get a Fresh View

- Easy to miss the cause of a problem when you have been looking too long.
  - Explain it to your lab partner
  - Explain it to your roommate
  - Take a shower
  - Go for a hike



If You Didn't Fix It, It Ain't Fixed

- Sometimes bugs will seem to go away even though you haven't done anything you thought should fix them.
- These bugs usually reappear
  - Often at the worst possible time

Debugging Down Under

Parting Words

- It is hubris to expect your program to work on the first try
- Be sure you have a clear idea of what it should do
  - Programming by trial and error is a recipe for slow progress
- Be able to recognize a malfunction
- If the malfunction is reproducible, add print statements and use divide-and-conquer
- If not, try to make it reproducible
- Pair programming can be a huge help